LA-UR-20-20260
Approved for public release; distribution is unlimited.

Title: SNNzkSNARK An Efficient Design and Implementation of a Secure Neural Network Verification System Using zkSNARKs

Author(s): DeStefano, Zachary Louis

Intended for: For presenting to other students at my university.

Issued: 2020-01-10

Los Alamos NATIONAL LABORATORY EST.1943

SNNzkSNARK

An Efficient Design and Implementation of a Secure Neural Network Verification System Using zkSNARKs

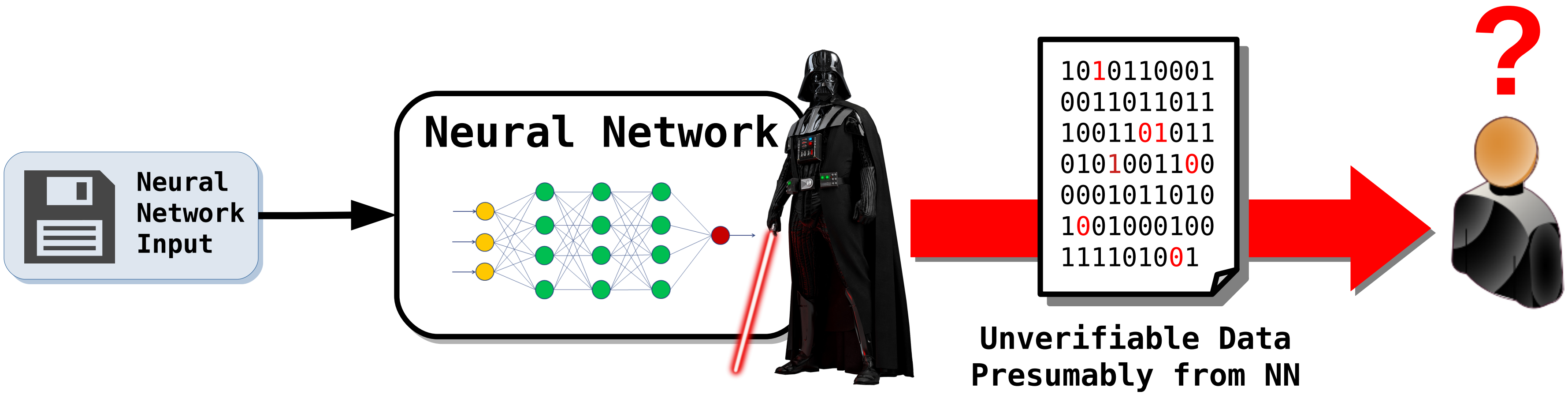
Zachary DeStefano

A-4: Advanced Research in Cyber Systems

Cyber Toaster - Research Track

UNCLASSIFIED

Current Problem



Challenges in Nuclear Verification

"...an **ever-increasing burden** is being placed on our nuclear safeguards **inspectors** and **analytical staff**..."

"...we have **increased** the number of **surveillance cameras** installed at facilities where nuclear material is present by a third since 2010 to nearly 1,600. The number of **unattended monitoring systems** has **risen** by 16 percent to 171, while the number of **remotely readable, tamper-proof seals** placed on nuclear material has jumped by nearly 280 percent to 560..."

"...we are approaching the **limits** of what is possible given the need to maintain a sufficient number of **inspectors** in the field..."

- Former IAEA Director General Yukiya Amano 04/05/2019

Nuclear Treaty Verification

- Currently requires the need to disclose sensitive data, but many parties are not willing to disclose sensitive data
- There has been an exponential increase in the amount of nuclear material and number of nuclear facilities in the past decade alone
- There is a clear need for secure, updatable, and robust new technologies for nuclear treaty verification

Other Examples

- Security camera auditing
  - currently requires the need to reveal footage
- Secure patient diagnosis
  - currently requires the need to disclose medical PII

Zero Knowledge Proofs (π)

- A **zero knowledge proof (π)** is a way to prove a claim without leaking details about why the claim is true
- For any **computable property** P, π says:

$$" \exists \vec{s}, \vec{\mathrm{in}} : \mathrm{P}(\vec{\mathrm{in}}, \vec{\mathrm{out}}, \vec{\mathrm{s}}) = \top "$$

- With **soundness**: $\varphi(\vec{x}) = \bot \Rightarrow Pr[Ver(\pi) = \top] \leq \mathrm{negl}(\vec{x})$
- And **completeness**: $\varphi(\vec{x}) = \top \Rightarrow Pr[Ver(\pi) = \top] = 1$

Zero Knowledge Proof Example

- How do I convince my blind friend that I can tell the difference between these two balls without revealing which one is red or which one is blue?

Zero Knowledge Proof Example Solution

- Step 1: my friend shows me one of them and I remember its color

- Step 2: my friend hides them and reveals one of them randomly to me

- Step 3: I tell him if it is the same one that he revealed to me before or a different one

≠

- Step 4: I remember the new color and return to step 2 until my friend is convinced

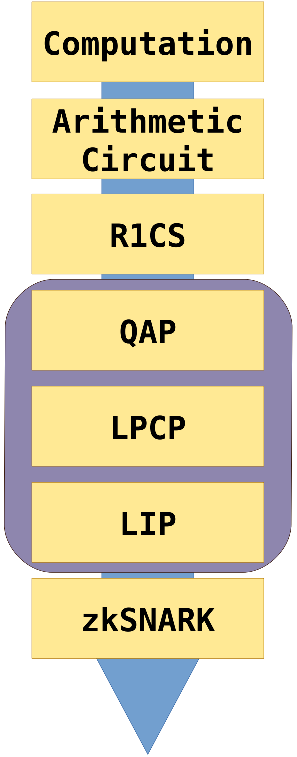
Zero Knowledge Proof Example Solution

- Every time my friend challenges me to identify if the ball was swapped, I have a 50% chance of getting it right just due to luck
- To convince him that I am not just very lucky, but indeed have secret information to help me distinguish the two balls, we need to repeat this challenge-response many times

Number of Rounds	Chance of getting all correct due merely to luck	Equivalent Odds
1	1/2	Guessing A Coin Toss Correctly
15	1/32,768	Being Dealt A Royal Flush In One Round Of Poker
33	1/8,589,934,592	Getting a Particular Person From A Random Sample
60	1/1,152,921,504,606,846,976	Getting a Particular Grain Of Sand On Earth From A Random Sample

zkSNARKS

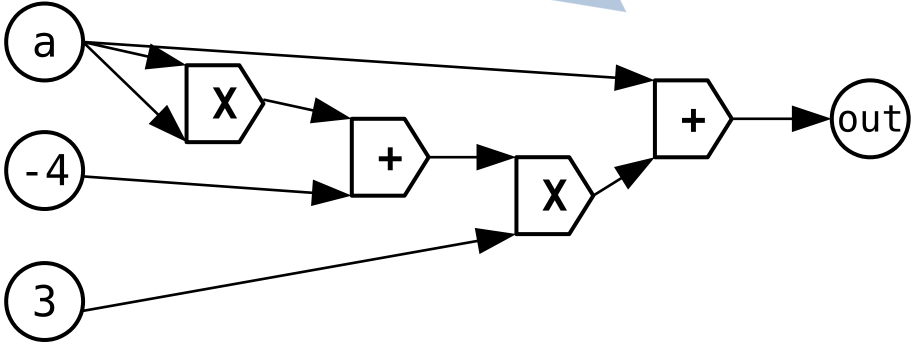
- **z**ero-**k**nowledge:
  - No secret information is revealed by the proof
- **S**uccinct:
  - The size of the proof that is generated is small
- **N**on-interactive:
  - no challenge-response protocol
- **AR**gument of **K**nowledge:
  - It is computationally intractable for the prover to produce a fake proof

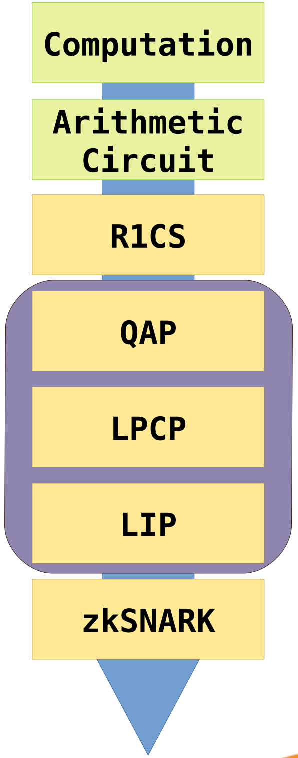


Computation to Arithmetic Circuit

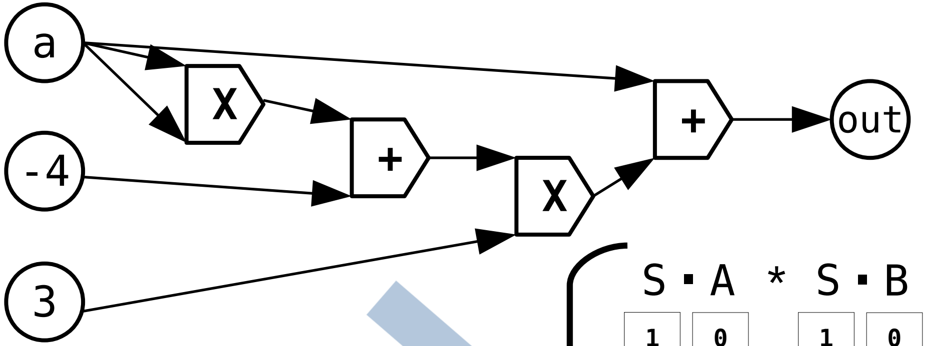
```
int secretFormula(int a){
     int b = a*a-4;
     return 3*b+a;
}
```

```
int secretFormula(int a){
     int t0  = a*a;
     int b   = t0+-4;
     int t1  = 3*b;
     int out = t1+a;
     return out;
}
```

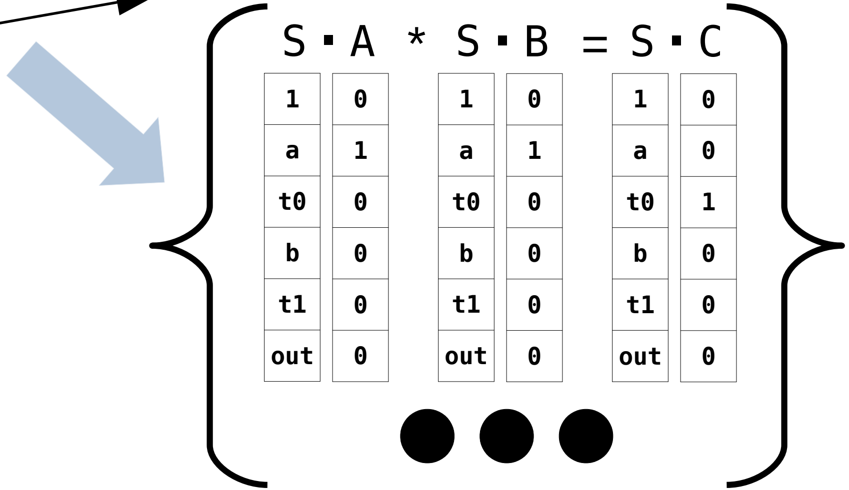


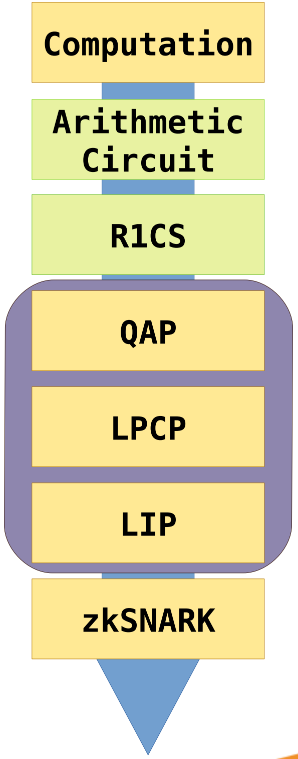


Arithmetic Circuit to R1CS




```
int t0  = a*a;
int b   = t0+-4;
int t1  = 3*b;
int out = t1+a;
```



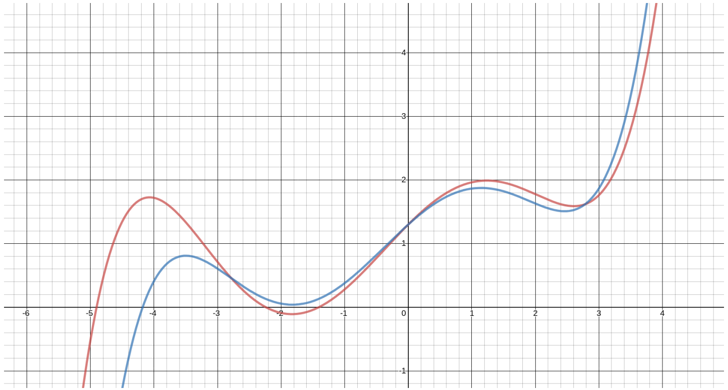


R1CS to QAP

S·A * S·B = S·C

1	0	1	0	1	0
a	1	a	1	a	0
t0	0	t0	0	t0	1
b	0	b	0	b	0
t1	0	t1	0	t1	0
out	0	out	0	out	0

● ● ●

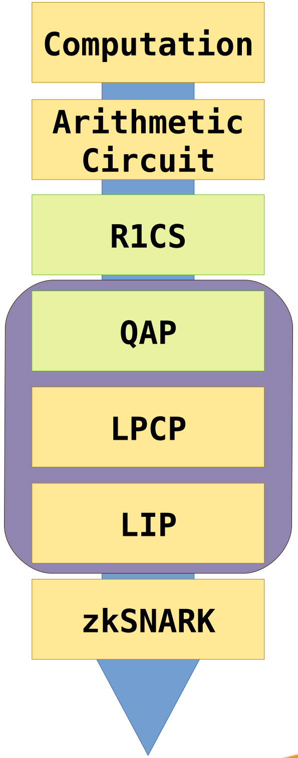


Polynomials of n degree intersect at most n number of times unless they are identical.

$$A(x) * B(x) - C(x) = Z(x)*H$$

$A_0(x)$	$B_0(x)$	$C_0(x)$	$Z_0(x)$
$A_1(x)$	$B_1(x)$	$C_1(x)$	$Z_1(x)$
$A_2(x)$	$B_2(x)$	$C_2(x)$	$Z_2(x)$
$A_3(x)$	$B_3(x)$	$C_3(x)$	$Z_3(x)$
$A_4(x)$	$B_4(x)$	$C_4(x)$	$Z_4(x)$
$A_5(x)$	$B_5(x)$	$C_5(x)$	$Z_5(x)$

The Zero Knowledge Is Added In This Step!

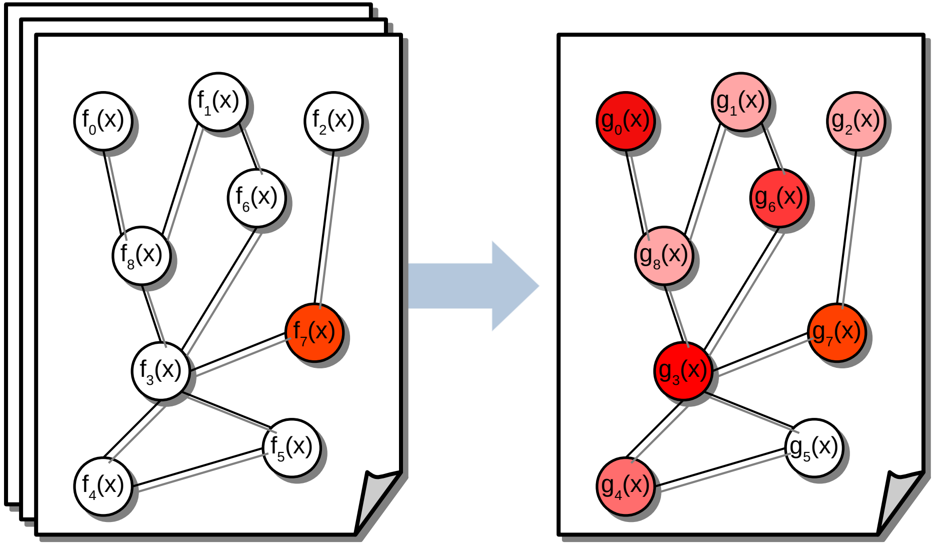


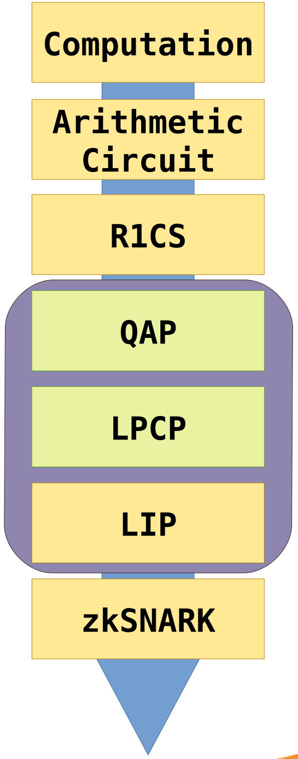
QAP to LPCP

$$A(x) * B(x) - C(x) = Z(x) * H$$

$A_0(x)$	$B_0(x)$	$C_0(x)$	$Z_0(x)$
$A_1(x)$	$B_1(x)$	$C_1(x)$	$Z_1(x)$
$A_2(x)$	$B_2(x)$	$C_2(x)$	$Z_2(x)$
$A_3(x)$	$B_3(x)$	$C_3(x)$	$Z_3(x)$
$A_4(x)$	$B_4(x)$	$C_4(x)$	$Z_4(x)$
$A_5(x)$	$B_5(x)$	$C_5(x)$	$Z_5(x)$

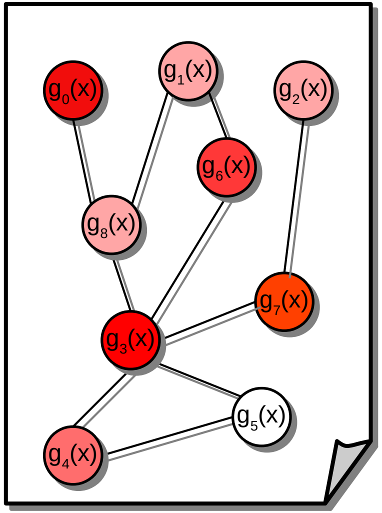
Round reduction
from O(poly(n)) to O(log(n)) rounds





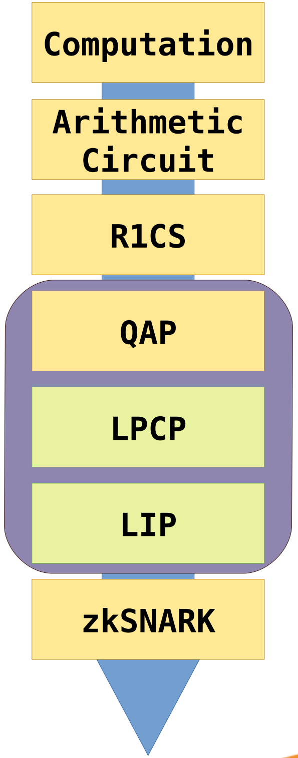
Succinctness Is Added In This Step!

LPCP to LIP



Round reduction from O(log(n)) to 2

$$\langle \pi, q_0 \rangle$$
$$\langle \pi, q_1 \rangle$$
$$\langle \pi, q_2 \rangle$$
$$\langle \pi, q_3 \rangle$$
$$\vdots$$
$$\langle \pi, q_m \rangle$$



Interactivity Is Reduced But Not Completely Removed In This Step!

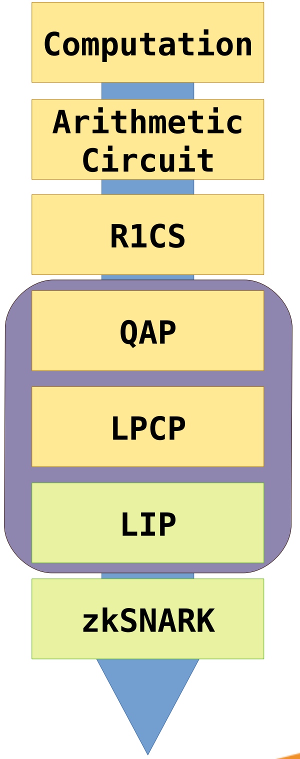
LIP to zkSNARK

$$\langle \pi, q_0 \rangle$$
$$\langle \pi, q_1 \rangle$$
$$\langle \pi, q_2 \rangle$$
$$\langle \pi, q_3 \rangle$$
$$\vdots$$
$$\langle \pi, q_m \rangle$$

Final round reduction from 2 to 1

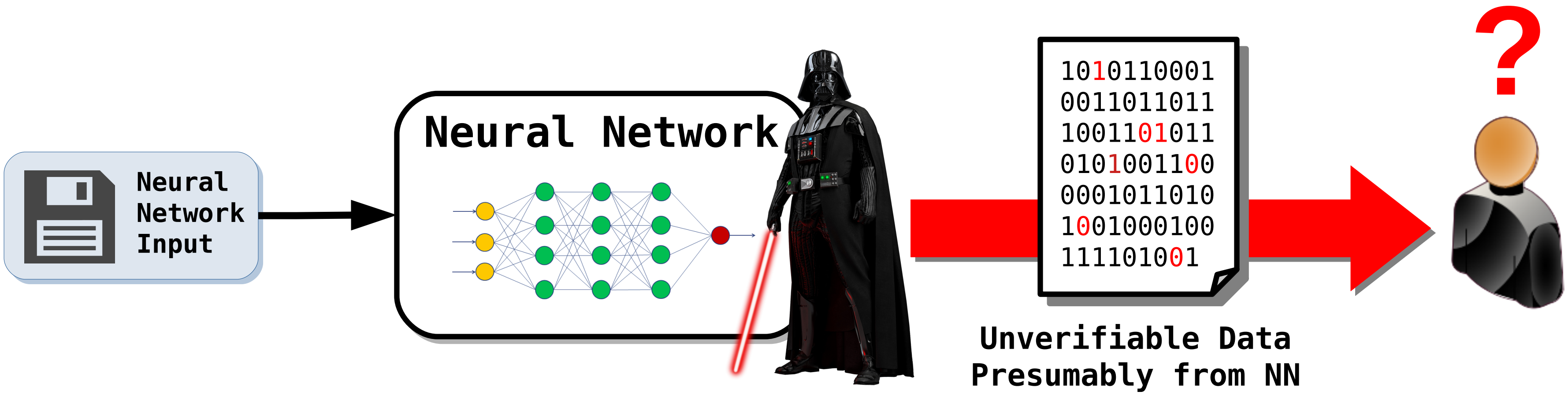
$$\pi$$

The zkProof is of constant size relative to the input

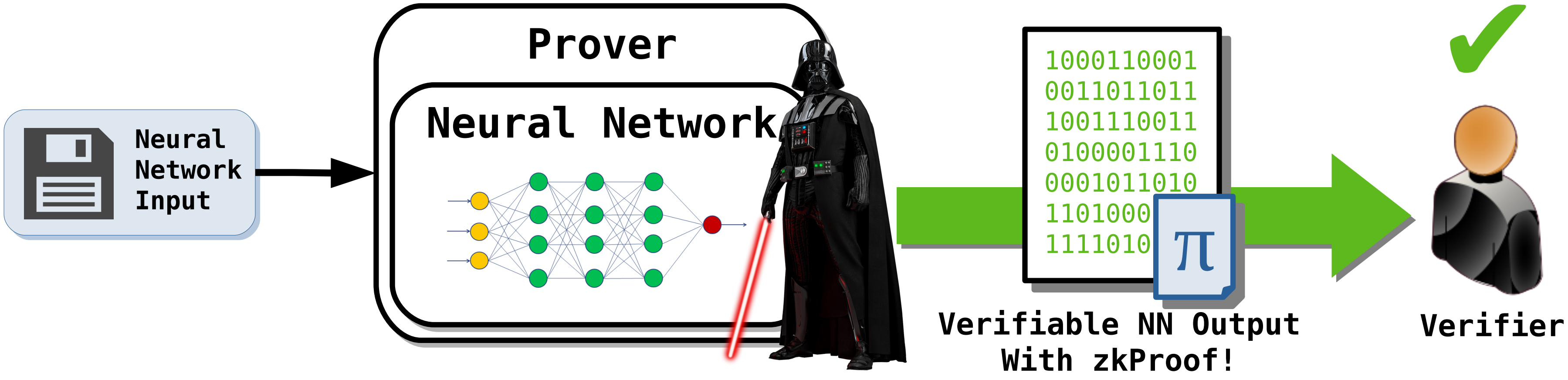


Interactivity Is Removed In This Step!

Current Problem

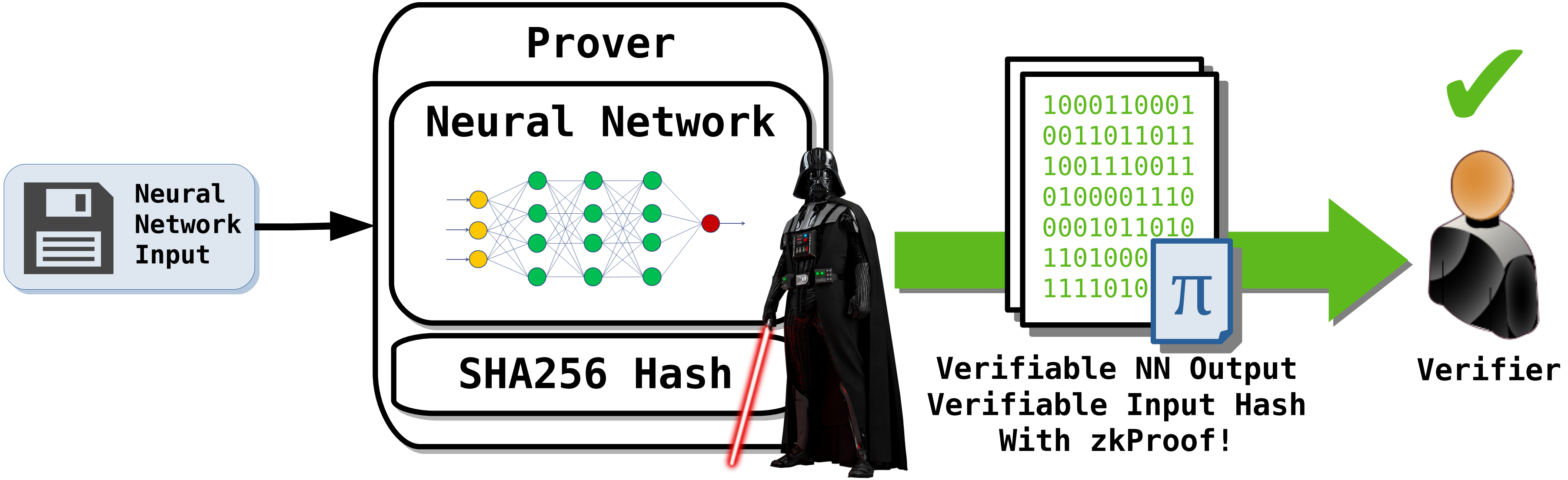


Solution Step 1



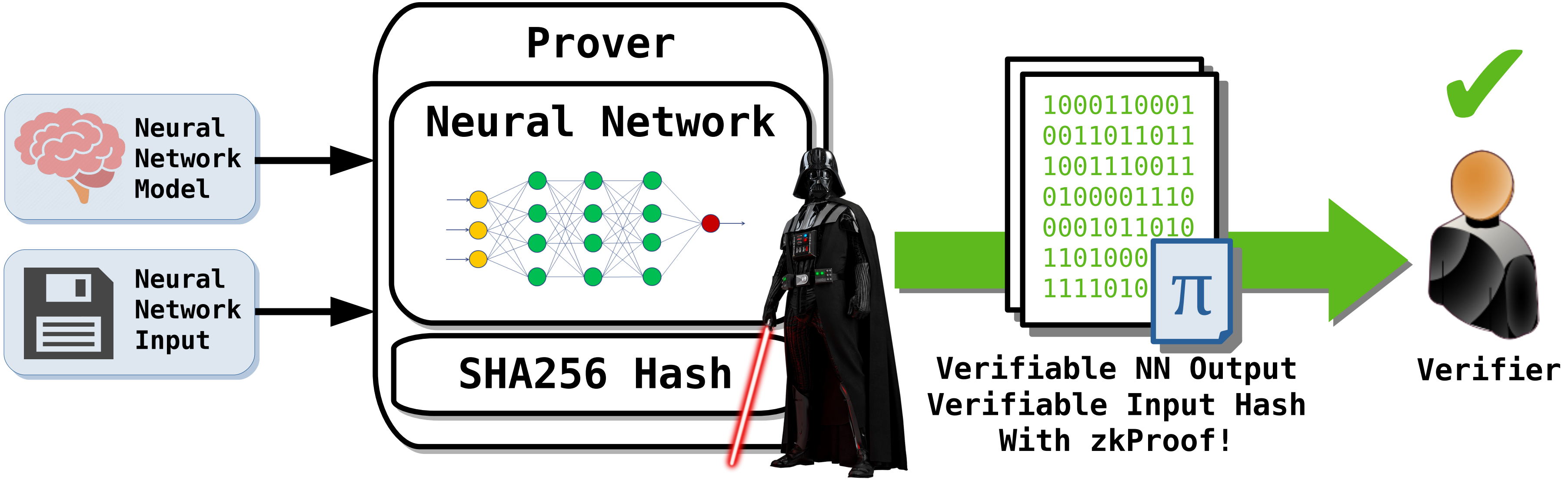
...but nothing is preventing the hidden input from being swapped to produce an undesirable result

Solution Step 2



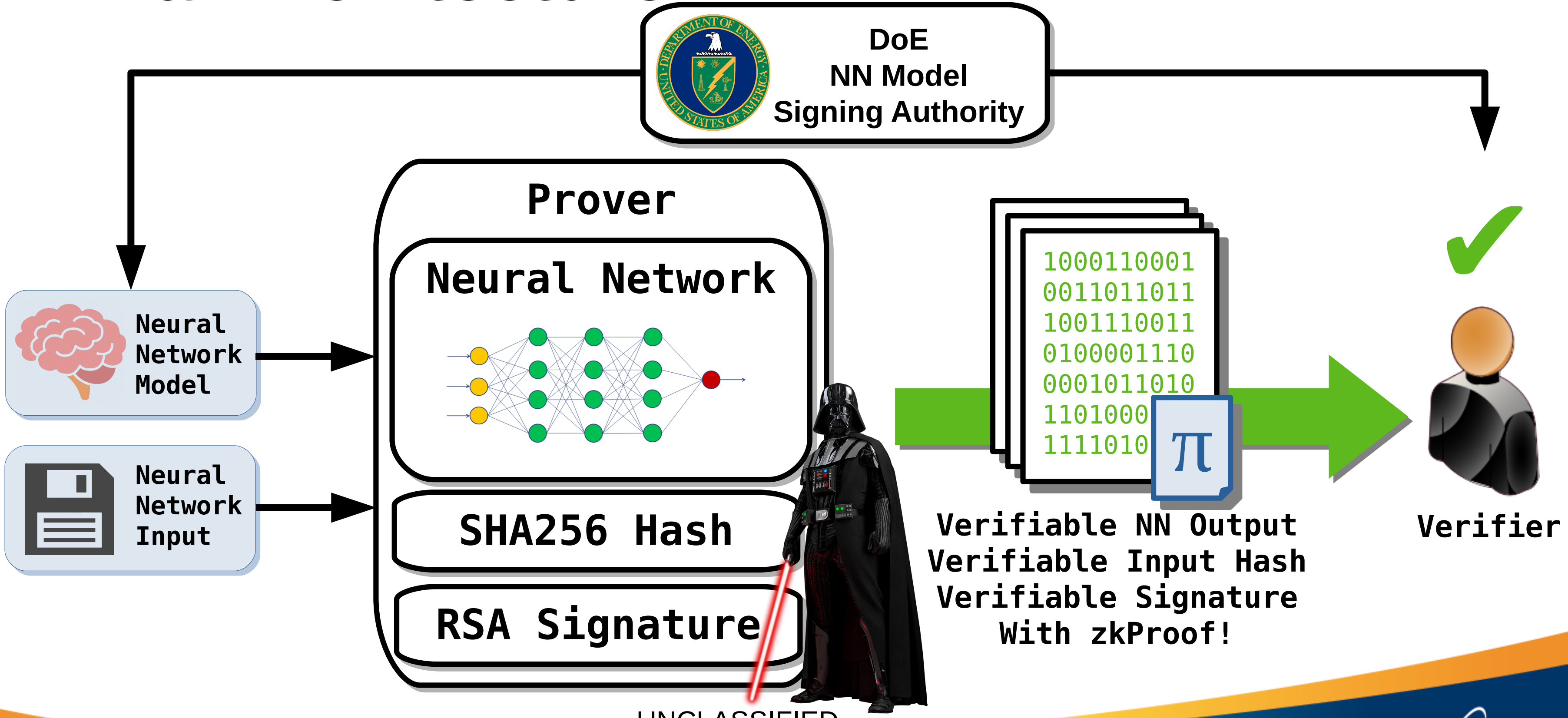
...but the neural network cannot be updated without compiling new verifier and prover applications

Solution Step 3



...but we have no way of knowing if the neural network parameters were tampered with or not

Final Architecture



MNIST Demo

Input Hash (32 Bytes):
0x0D5394498EC5602F7D29DEC1A114CB39

Model Signature (256 Bytes):
A6 2E 94 84 6B 41 8F 69 89 34 E4 92 ... 9F 45

Output Weights (42 Bytes):
0:(0.00000) 1:(0.03225) 2:(0.23016) 3:(0.00000)
4:(0.00011) 5:(0.00078) 6:(0.00000) 7:(0.99761)
8:(0.03148) 9:(0.00000)

ZK Proof (~50KB):
11011100 11000101 00100101 01100111...00001010

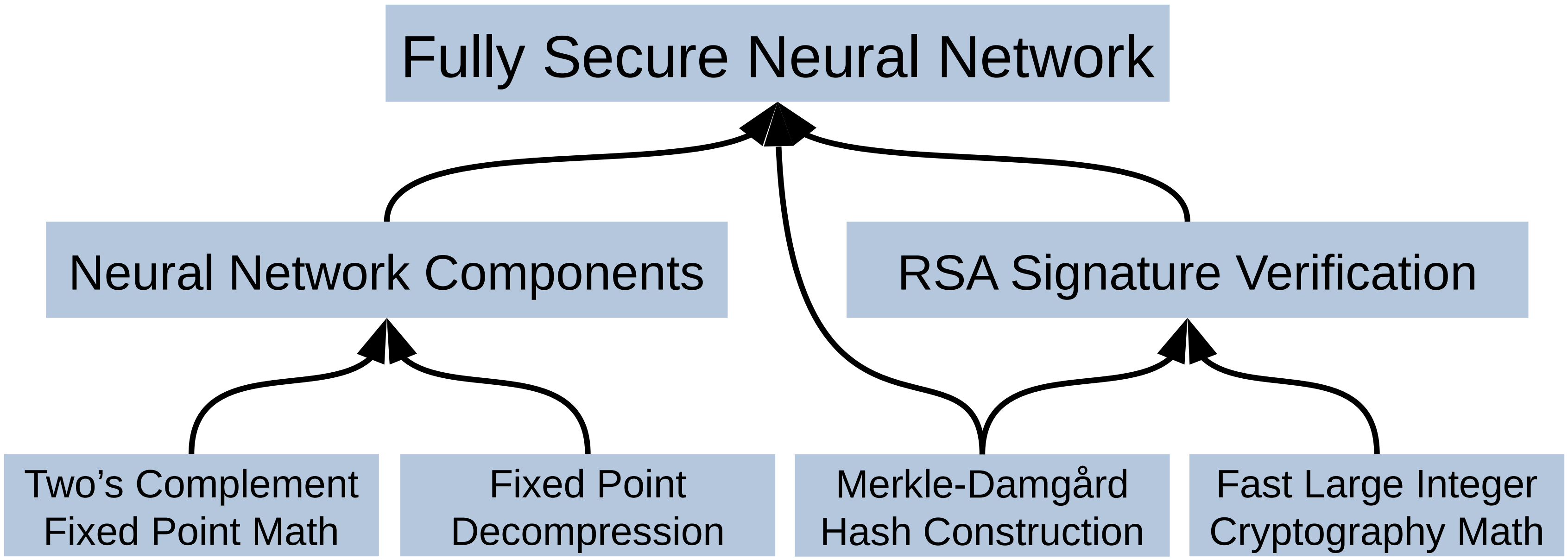
(ZK Proof size is equal to the input size)

Secret Input:

Performance Metrics

MNIST Demo Stats	Setup Time (128 Threads)	Setup Ram	Prover Time	Proving Key Size
With RSA	5-10 min	80 GB	>1 min	30 GB
Without RSA	3-5 min	12 GB	<10 seconds	2 GB

Our Gadget Hierarchy



Future Work

- Implement additional neural network features which effectively leverage the structure of R1CS constraints, such as convolutions.
- Implement a neural network specific compiler and optimizer for the reduction from Computation to R1CS
- Develop a method that is more efficient than QAPs or SSPs for the expansion of R1CS to a Linear PCP
- Develop and Implement a post-quantum secure zkSNARK construction scheme and implement post-quantum cryptographic gadgets in R1CS

Questions?